

We claim:

1. An HDL simulator comprising: an automated interface to compiled or interpreted application code written in a general purpose programming language.
5
2. A simulator as in claim 1 using either compiled or interpreted HDL code which may use special-purpose hardware to accelerate the simulation.
3. A simulator as in claim 1 wherein the interface enables the HDL code to have a direct data access to and from the application code.
10
4. A simulator as in claim 3 which automatically maps and converts HDL data types to and from programming language data types.
- 15 5. A simulator as in claim 4 in which said data type mapping may be applied either to the arguments of routine calls or to the direct data access.
6. A simulator as in claim 4 further comprising a programming language calling mechanism and using said automatic data type mapping of arguments, which enables the HDL to call application
20 code routines compiled with a standard compiler, and enables such routines to call functions in the HDL.
7. A simulator as in claim 6 which automatically generates wrappers for the interface.
- 25 8. A simulator as in claim 7 wherein the wrappers also automatically map data types for direct data access when the application code is compiled.
9. A simulator as in claim 7 wherein the wrappers can output a message upon the occurrence of a call or return.

10. A simulator as in claim 7 in which said automatically generated wrappers include automatic threading which enables compiled application code to call tasks in the HDL.
- 5 11. A simulator as in claim 3 further comprising a programming language calling mechanism which enables the HDL to call application code routines compiled with a standard compiler, and enables such routines to call functions in the HDL.
- 10 12. A simulator as in claim 11 which automatically generates wrappers for the interface.
- 15 13. A simulator as in claim 12 wherein the wrappers can output a message upon the occurrence of a call or return.
14. A simulator as in claim 12 in which said automatically generated wrappers include automatic threading which enables compiled application code to call tasks in the HDL.
- 15 15. A simulator as in claim 3 which automatically generates wrappers for the interface.
- 20 16. A simulator as in claim 15 wherein the wrappers can output a message upon the occurrence of a call or return.
- 20 17. A simulator as in claim 15 in which said automatically generated wrappers include automatic threading which enables compiled application code to call tasks in the HDL.
- 25 18. A simulator as in claim 1 with automatic data type mapping.
- 25 19. A simulator as in claim 18 which applies a programming language calling mechanism to the HDL.
- 20 20. A simulator as in claim 19 which automatically generates wrappers for the interface.
- 30 21. A simulator as in claim 20 with automatic threading.

22. A simulator as in claim 18 which automatically generates wrappers for the interface.

23. A simulator as in claim 22 with automatic threading.

5

24. A mixed language simulator comprising: an HDL simulator including an automated interface to compiled or interpreted application code written in a general purpose programming language.

25. A mixed language simulator comprising: HDL simulator means; and

10 an automated interface to application code.